



사용자 입력 기반 건강식품 추천 챗봇

Subtitle

AI 헬스 메이트 챗봇

WHY?



A

건강 식품에 대한 기호 증가

건강에 대한 경각심과 관심이 점점 늘어나는 추이

B

수 많은 제품 및 정보

다양한 기업에서 출시하는 제품의 수와 정보가 많음

C

사용자 니즈에 맞는 제품 찾기 힘들

수 많은 제품과 정보로 인해 사용자에게 맞는 제품을 찾기가 어려워지고 있는 상황

D



문제점을 극복할 기획 발표

기존에 달렸던 리뷰를 데이터화 + 건강 식품과 관련한 전문가 의견을 취합해 사용자에게 맞는 건강 식품을 추천해주는 챗봇을 개발해 보기로 기획

CONTENTS

1

데이터 수집



2

데이터 분석



3

모델링



4

RAG 구현



5

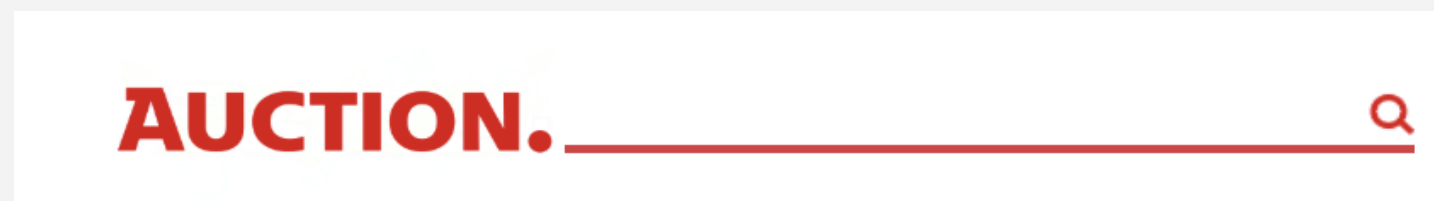
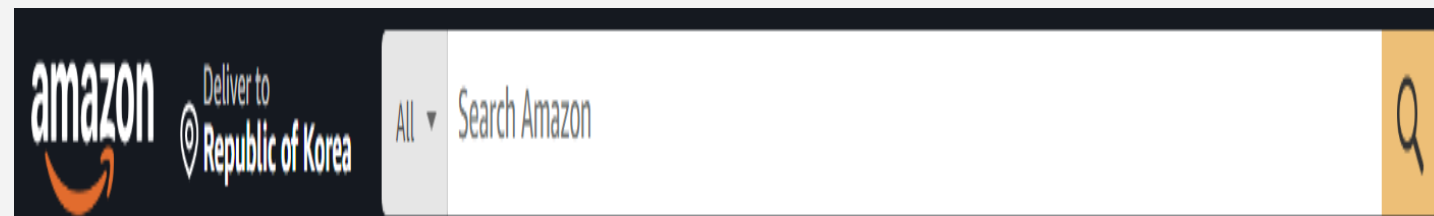
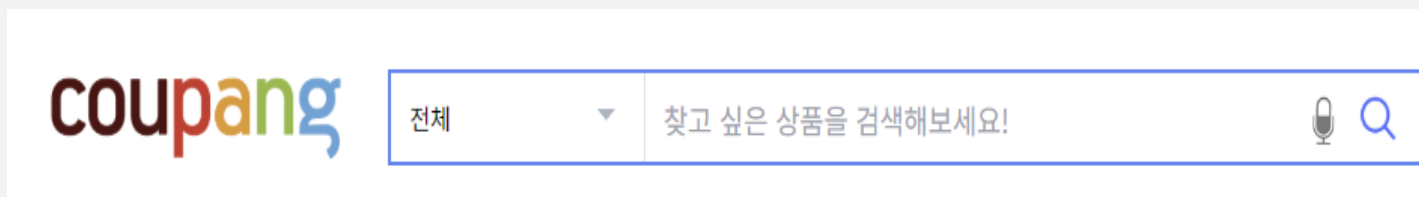
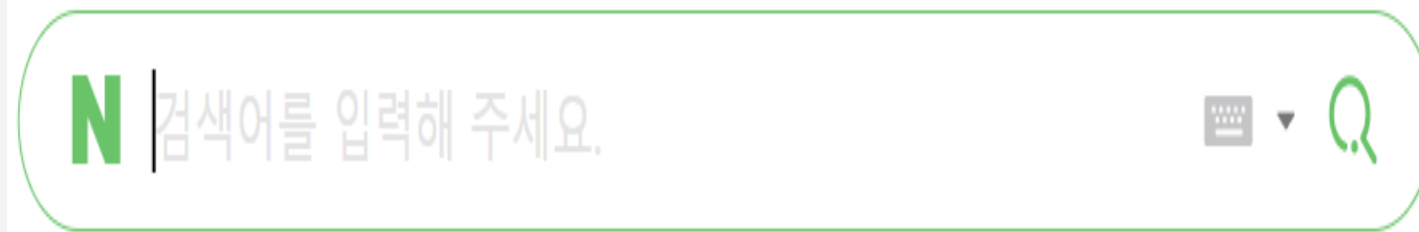
웹 페이지 구현



Chapter 1

데이터 수집

데이터 수집



제품 정보 및 리뷰 데이터 수집

- 단백질 셰이크, 아연, 오메가3, 유산균, 멀티비타민 각각 제품 30개씩 수집
- 제품 정보 및 제품 당 리뷰 최대 500개 추출
- 수집 플랫폼: 네이버스토어, 쿠팡, 아마존, 옥션

선택 이유

A 네이버 스마트 스토어

- 국내 이커머스 시장에 큰 비율 차지
- 다양한 연령대와 소비 패턴에 대한 리뷰 확보 가능

B 아마존

- 세계적으로 방대한 리뷰와 상품을 보유하고 있음
- 국제적으로 다양한 연령대와 취향에 따른 리뷰 확보 가능

C 쿠팡

- 국내에서 가장 인기 있는 플랫폼
- 쿠팡 서포터즈로 인한 구체적이고 체계적인 리뷰

D 옥션

- 상대적으로 마케팅 비중이 적은 플랫폼의 리뷰를 수집해 데이터의 신뢰도 향상

Chapter 2

데이터 분석

분석 기준 세부사항



카테고리

1

건강기능식품 카테고리별 리뷰 분석



감성별

2

긍정/부정 리뷰에서 자주 언급되는 키워드 비교

분석 과정

A 분석의 초기와 한계점

- 워드 클라우드 시도
- 유의미한 키워드 도출이 어렵다는 한계점 발견

B 자연어 처리 도입

- 이를 보완하기 위해 자연어 처리 기술 도입
- 복합적인 의미 도출 기대

C 데이터 준비 및 분석

- 전처리 된 csv 파일 이용
- 허깅 페이스 모델을 활용해 감성 분석 및 리뷰 요약 진행
- 새 csv 생성

D 최종 인사이트 도출

- TF-IDF와 제미니AI 활용
- 카테고리 속 긍/부정 감성 분석을 통해 인사이트 도출
- 이를 바탕으로 키워드 추출

인사이트 도출

각 카테고리별 긍정과 부정이 높은 비율의 리뷰를 분석해 소비자의 니즈 및 전체적인 인사이트를 정리.

A

단백질

맛, 용량, 섭취 용이성, 재구매 의사 등 다양한 측면에서 긍/부정 반응이 공존, 맛과 섭취 용이성을 중요하게 평가

C

유산균

장 건강, 소화 개선과 같이 효과에 대한 니즈가 상당히 높은 편, 따라서 부작용에 대한 평가도 공존
맛에 대한 평가는 미미한 수준

E

오메가3

냄새, 섭취 용이성 측면에서 긍/부정 평가가 공존, 알약 크기 및 효과, 냄새 여부 등이 소비자 만족도에 큰 영향을 줌

B

비타민

알약 크기와 함께 브랜드에 대한 신뢰도를 중요하게 여김, 맛과 냄새, 그리고 부작용에 대한 부정적 평가 다수

D

아연

주로 효과에 대해서 긍정적 평가와 부정적 평가가 공존, 알약 크기 또한 소비자 만족도에 차이에 큰 영향을 줌

최종 키워드

A

단백질

맛, 대용량, 목 넘김이 편한, 가성비 좋은,
근육 강화

B

비타민

알약 크기가 작은, 부작용 없는, 고함량,
믿을 수 있는 브랜드

C

유산균

장 건강, 변비 해소, 소화 개선, 활력 증진

D

아연

면역 체계에 좋은, 삼키기 쉬운, 섭취량이
적당한, 부작용 없는

E

오메가3

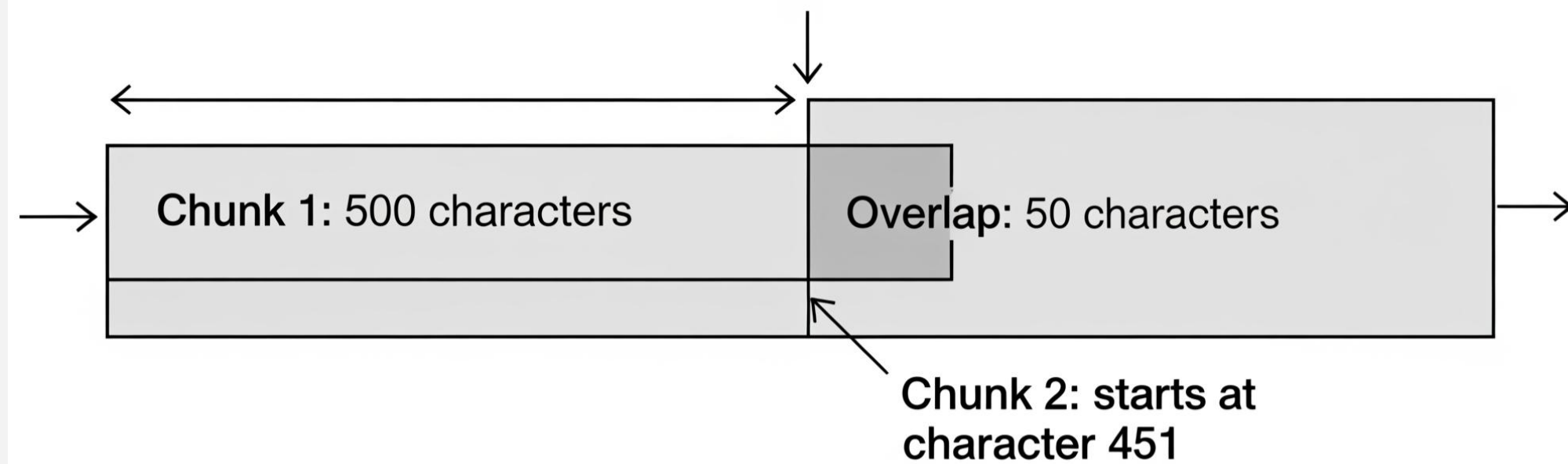
심장 건강에 좋은, 관절에 좋은, 생선
맛(향)이 나지 않는, 삼키기 쉬운, 부작용
없는

Chapter 3

모델링



Chunking



- Chunk_size = 500 으로 설정하여 500글자 단위로 문서 분할
- Chunk_overlap = 50으로 설정하여 문서의 앞 뒤 50글자는 다음 문서와 겹치게 설정하여 정보 연속성 및 문맥 손실방지

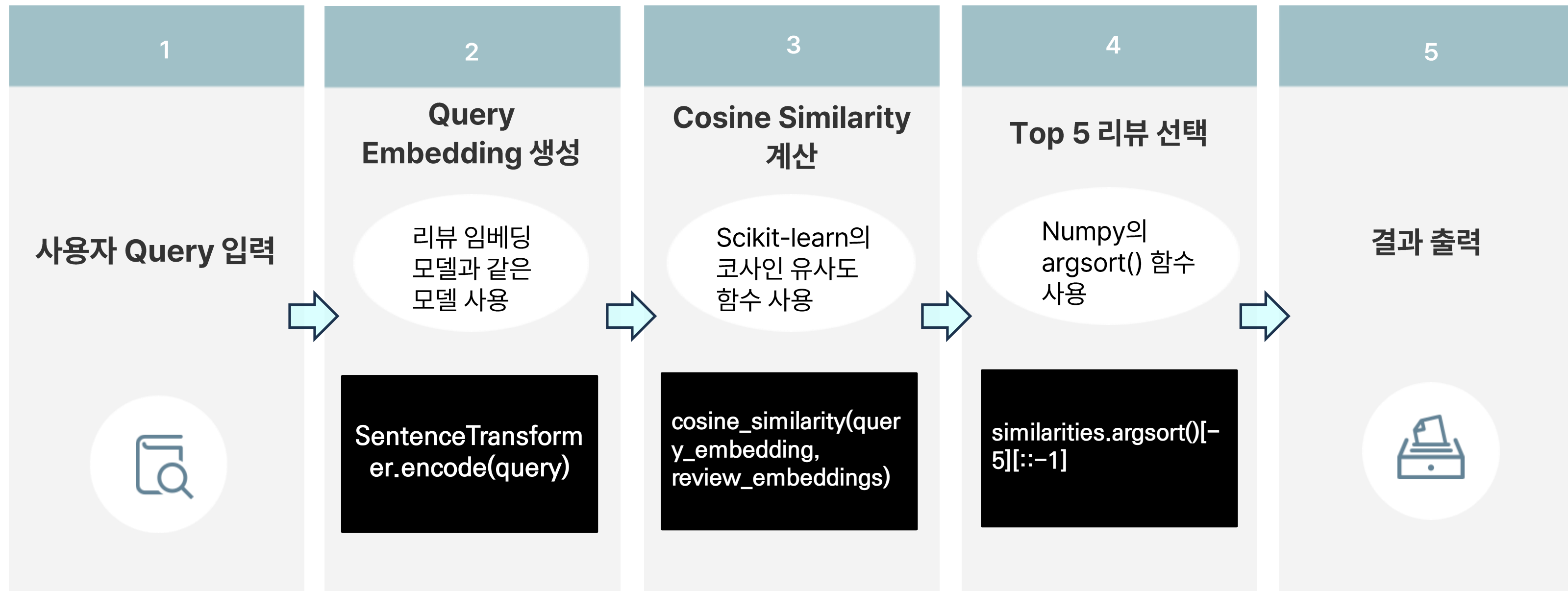
Embedding

original_review_index	chunk_text	embedding
0	0 초코맛 사려다가 조금 더 저렴해서 샀는데 생각보다 맛있습니...	[-0.03533213213086128, -0.16412071883678436, 0...
1	1 맛있고 먹기 좋습니다	[0.2144167572259903, 0.1854843646287918, 0.042...
2	2 운동 전후 먹으면 든든하고 좋습니다.	[0.20139068365097046, -0.03795986995100975, -0...
3	3 양이 많아서 좋아요 맛도 있어요.	[0.19032299518585205, -0.18672198057174683, 0....
4	4 만족합니다 감사합니다	[0.12037263065576553, 0.18490718305110931, 0.1...
5	5 맛있어요 효과 좋아요	[0.13925422728061676, 0.04154832661151886, 0.0...
6	6 가성비 최고인듯 매우 만족합니다	[0.14988018572330475, 0.13022376596927643, -0....
7	7 유당불내증 있어서 시켰는데 속 편하고 좋아요. 다음에도 시킬것 같아요. ㅋㅋ	[0.058354321867227554, 0.11737152934074402, 0....
8	8 배송빠르고좋아요!!!!!!	[0.10080527514219284, 0.17413099110126495, 0.1...
9	9 잘먹겠습니다 배송도 적당히 잘 왔습니다	[0.17237910628318787, 0.10714202374219894, 0.0...

각 Chunk를
SentenceTransformer
모델로 벡터화

임베딩 모델 : HuggingFace의
다국어 모델 사용

Cosine Similarity



코사인 유사도의 한계

A

문맥 미반영 문제

예)

사용자 쿼리 : '피로 회복'

유사도 결과 상위 리뷰 中 :

- '피로 회복에 도움이 되는지는 잘 모르겠다'
- '피로 회복 목적으로 샀는데 효과는 좀 더 지켜봐야 할 거 같다'

B

비효율적

멘토링 피드백 :

사용자 쿼리를 모든

리뷰(162,574개)에 대해서 코사인
유사도 측정을 하는 것은 비효율적

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 162574 entries, 0 to 162573
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   고유번호    161074 non-null object
 1   제품명      162574 non-null object
 2   리뷰        162462 non-null object
 3   상세정보    141336 non-null object
 4   카테고리    162574 non-null object
dtypes: object(5)
memory usage: 6.2+ MB
```


문제 해결

1) '고유번호'로 그룹화

2) '제품명', '상세정보', '카테고리'에 대한 정보는 하나만 필요함으로
그룹 내에서 첫번째 값만 뽑음

3) '리뷰'는 서로 다른 리뷰 간에 '/'로 구분하여 하나로 묶음

```
df_grouped = (  
    df.groupby('고유번호')  
        .agg({  
            '제품명': 'first',  
            '상세정보': 'first',  
            '카테고리': 'first',  
            '리뷰': lambda x: ' / '.join(x.fillna('').astype(str))  
        })  
        .reset_index()  
)
```

(수정 전)하나의 리뷰 당 한 행
→ (수정 후)하나의 제품 당 한 행

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 521 entries, 0 to 520  
Data columns (total 5 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   고유번호    521 non-null    object  
1   제품명      521 non-null    object  
2   카테고리    521 non-null    object  
3   상세정보    458 non-null    object  
4   리뷰        520 non-null    object  
dtypes: object(5)  
memory usage: 20.5+ KB
```

문제 해결

GCP gemini API를 이용한 감성별 리뷰 요약

모델: gemini-2.5-pro 사용

긍정 리뷰만 임베딩하여 사용자 쿼리와 긍정 리뷰와의 코사인 유사도 비교

부정 리뷰와 중립 리뷰는
참고용으로 제시하기로 함

고유번호	제품명	카 테 고 리	상세정보	긍정리뷰	부정리뷰	중립리뷰	긍정_청크	긍정리뷰_임베딩	
0	10067386235	Newt365 유산균	NaN	섭취 후 변비가 해소되어 매일 화장실을 가게 되었다는 등 배변 활동에 대한 긍정적 ...	광고와 달리 효과를 전혀 보지 못했거나, 처음에는 효과가 있다가 사라졌다는 의견이 ...	제품을 막 받아서 아직 복용 전이거나, 복용한 지 얼마 되지 않아 효과를 기대하며 ...	섭취 후 변비가 해소되어 매일 화장실을 가게 되었다는 등 배변 활동에 대한 긍정적 ...	[0.10404543578624725, -0.11060380935668945, -0...	
1	10102052354	Sentroom_man	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명 (식약처고시) 비타민A, 비타민D, ...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 복부 팽만감과...	제품 효과에 대해 아직 잘 모르겠다는 의견이 다수 있었습니다. 꾸준히 복용하고 있지...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	[-0.06720072776079178, 0.08641089498996735, -0...
2	10102052354	Sentroom_man	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명 (식약처고시) 비타민A, 비타민D, ...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 복부 팽만감과...	제품 효과에 대해 아직 잘 모르겠다는 의견이 다수 있었습니다. 꾸준히 복용하고 있지...	저렴한 가격과 브랜드에 대한 신뢰도 긍정적인 요인으로 언급되었습니다.	[-0.2792225182056427, 0.12212540209293365, -0...

원하는 건강기능식품 종류를 입력해주세요.

건강기능식품 종류: 아연, 비타민, 유산균, 오메가3, 단백질 中 유산균
사용자 프롬프트: 소화 개선

인덱스: 54 | 유사도: 0.5732 | 제품명: MaryRuth Organics USDA Organic Liquid Probiotic | Digestive Health | Gut Health
긍정 리뷰: 소비자들은 소화 개선, 복부 팽만감 감소, 변비 해결 등 장 건강 문제에 대한 탁월한 효과에 매우 만족했습니다.

인덱스: 51 | 유사도: 0.5413 | 제품명: Garden of Life Once Daily Dr. Formulated Probiotics for Women 50 Billion CFU 16
긍정 리뷰: 소비자들은 소화 개선, 복부 팽만감 감소, 규칙적인 배변 활동에 큰 효과를 보았다고 평가합니다. 면역력 증진, 질

Chapter 4

RAG 구현

A

Lang Chain 기반

B

Llama Index 기반

Lang Chain 기반 RAG



A

문서 로드 & 인덱싱

수집한 PDF문서, 리뷰를 읽어 텍스트화
BM25 토큰 인덱스, FAISS 인덱스 구축

B

하이브리드 검색 & 융합

BM25 + Dense를 RRF로 융합
후보 문서 선별

C

리랭킹 & 컨텍스트 추출

CrossEncoder로 관련도 높은 순으로 재정렬
Top-K 문서 발췌, LLM 입력 컨텍스트 생성

D



LLM 응답 생성

skt/A.X-4.0-Light 모델로 JSON 형태 출력
최종 한국어 랜더링
(사용자 입력 / 추천 제품 / 전문가의 의견 / 근거 / 참고 문서)

Lang Chain 기반 RAG

```
# 청크 분할기
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size = 500,
    chunk_overlap = 50,
)

# PDF문서 로드 & 청크 분할
all_docs = []

for pdf_file in tqdm(pdf_files, desc = "PDF Load :"):
    loader = PyMuPDFLoader(pdf_file)
    docs = loader.load()
    all_docs.extend(docs)

all_chunks = text_splitter.split_documents(all_docs)

print(f"총 문서 수: {len(all_chunks)}")
```

- Langchain의 RecursiveCharacterTextSplitter를 사용하여 청크 분할기 생성
- PyMuPDFLoader로 PDF 문서 로드
- 청크 분할기로 PDF문서에서 수집된 텍스트 분할

Lang Chain 기반 RAG

```
def dense_search(query: str, top_k=40) -> List[Tuple[int, float]]:
    if not docs:
        return []
    q = emb_model_doc.encode([query], convert_to_numpy=True)[0]
    q = q / (np.linalg.norm(q) + 1e-12)
    if index is not None and use_faiss:
        D, I = index.search(q[np.newaxis, :].astype('float32'), top_k)
        return [(int(i), float(d)) for i, d in zip(I[0], D[0])]
    sims = (doc_embeddings @ q) / (np.linalg.norm(doc_embeddings, axis=1) + 1e-12)
    top_idx = np.argsort(-sims)[:top_k]
    return [(int(i), float(sims[i])) for i in top_idx]
```

```
def sparse_search(query: str, top_k=80) -> List[Tuple[int, float]]:
    if not docs or bm25 is None:
        return []
    tokens = simple_tokenize_ko(query)
    scores = bm25.get_scores(tokens)
    top_idx = np.argsort(-scores)[:top_k]
    return [(int(i), float(scores[i])) for i in top_idx]
```

- BM25(Sparse Search) : 단어 기반
통계적 검색 -> 키워드 매칭 강점
- Dense Retrieval : 의미 기반 임베딩 검색

Lang Chain 기반 RAG

```
def rrf_fuse(dense: List[Tuple[int, float]], sparse: List[Tuple[int, float]], k: int = 60, top_k: int = 50):
    ranks: Dict[int, float] = {}
    for lst in [dense, sparse]:
        for rank, (idx, _) in enumerate(lst):
            ranks[idx] = ranks.get(idx, 0.0) + 1.0 / (k + rank + 1)
    fused = sorted(ranks.items(), key=lambda x: -x[1])[:top_k]
    return [idx for idx, _ in fused]
```

```
def rerank(query: str, candidate_ids: List[int], top_k=10) -> List[int]:
    if not candidate_ids:
        return []
    pairs = [[query, docs[i]["text"]] for i in candidate_ids]
    if reranker:
        scores = reranker.predict(pairs)
    else:
        qset = set(simple_tokenize_ko(query))
        scores = [len(qset & set(simple_tokenize_ko(docs[i]["text"]))) for i in candidate_ids]
    order = np.argsort(-np.array(scores))[:top_k]
    return [candidate_ids[i] for i in order]
```

- RRF(Reciprocal Rank Fusion) :

두 방식의 장점 융합

- Reranking :

최종 후보 문서 재정렬

RAG 출력 결과

질문 : 60대 남자 건강식품 추천

사용자 입력 : 60대 남자 건강식품 추천

추천 제품 : 오메가-3

전문가의 의견 : 60대 남성의 경우 심혈관 건강 유지를 위해 오메가-3 지방산 섭취가 권장됩니다.

근거 : 심혈관 질환 예방을 위한 오메가-3의 역할; 제공된 문서의 관련 페이지

참고 문서 : 노인 영양 관리 지침 p.5(doc-05)

- 질문 : 사용자의 Input
- 전문가의 의견, 근거 : 수집한 논문(PDF형식)을 기반으로 한 Output 출력

RAG 한계

질문 : 60대 남자 건강식품 추천

사용자 입력 : 60대 남자 건강식품 추천

추천 제품 : 오메가-3

전문가의 의견 : 60대 남성의 경우 심혈관 건강 유지를 위해 오메가-3 지방산 섭취가 권장됩니다.

근거 : 심혈관 질환 예방을 위한 오메가-3의 역할; 제공된 문서의 관련 페이지

참고 문서 : 노인 영양 관리 지침 p.5(doc-05)

한계점 :

- 수집한 PDF 논문의 내용이 학술적인 내용이고 사용자의 Input은 일반적으로 많이 사용하는 내용이기 때문에 RAG가 PDF를 참조하지 못함
- 따라서 RAG의 환각으로 인해 거짓 정보를 제공

Llama Index 기반 RAG

```
### SentenceSplitter

text_splitter = SentenceSplitter(
    chunk_size=200,
    chunk_overlap=50
)
```

1
Text_splitter

2
Embedding

```
### embedding

# 임베딩 모델 생성
embed_model = HuggingFaceEmbedding(
    model_name = 'paraphrase-multilingual-MiniLM-L12-v2',
    device='cpu' # gpu가 있다면 'cuda'
)
```

3
LLM

4
Settings

```
# llm 생성
llm = GoogleGenAI(
    model='gemini-2.5-pro',
    request_timeout=120.0,
    temperature=0.0,
    api_key=google_api_key
)
```

```
### settings

# llm, embedding, text_splitter 모델 설정
Settings.llm = llm
Settings.embed_model = embed_model
Settings.text_splitter = text_splitter
```

Llama Index 기반 RAG

RAG에서 참조할 텍스트 파일 로드

5

```
# 문서 로딩
txt_documents = SimpleDirectoryReader(input_files=files).load_data()
```

Index 생성

```
txt_index = VectorStoreIndex.from_documents(txt_documents)
```

Query_engine 생성

6

```
# query_engine 생성
txt_engine = txt_index.as_query_engine(similarity_top_k=10, include_metadata=True)
```

Tool 생성

7

```
# pdf_tool 생성
txt_tool = QueryEngineTool(
    query_engine=txt_engine,
    metadata=ToolMetadata(
        name='txt_tool',
        description='건강기능식품에 들어간 원료의 효능을 알려주는 도구입니다'
    )
)
```

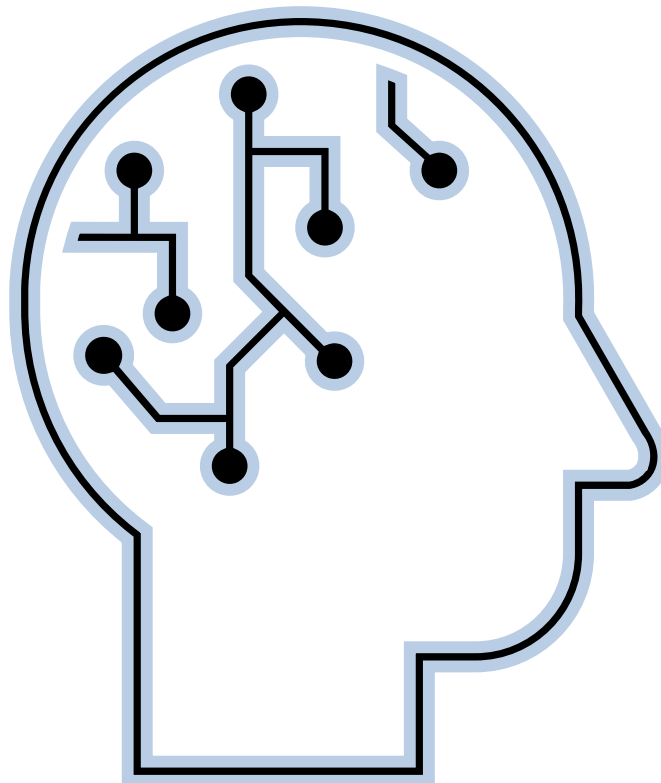
Agent 생성

8

```
# 시스템 프롬프트 정의
react_system_prompt = f"""
당신은 건강기능식품 AI 비서입니다. txt_tool을 반드시 활용하세요.
"""

# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[txt_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)
```

RAG 시행착오



A

프롬프트 문제

프롬프트 입력 위치를 system_prompt → user_msg 로 변경

B

Tool 문제 1

Product_tool 삭제, pdf_tool 유지

C

Tool 문제 2

Tool을 활용 못하는 원인 파악
→ Tool에 들어가는 내용 변경

D



문제 해결

수정 후 프롬프트 작성 대로 출력됨

프롬프트 문제

Before

agent 생성시 ReActAgent의 매개변수인
system_prompt에 프롬프트를 다 넣음
→ 프롬프트를 전혀 반영 못하고 출력

After

agent 실행시 .run()의 매개변수인 user_msg에 전달
→ 개선된 점: 수정 후 프롬프트 지시사항 대로 출력됨

```
# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[product_tool, txt_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)
```

```
response = await healthcare_agent.run(user_msg=f"""당신은 사용자의 요구에 가장 알맞는 제품을 추천하는 친절하고 유능한 건강기능식품 AI 비서입니다.
주어진 도구를 사용하여 질문에 대한 정보를 찾아야 합니다. 'pdf_tool'를 사용하세요.
'product_text'에 있는 상세정보에 있는 영양소 원료명을 'pdf_tool'을 참조하여 제일 사용자의 요구에 맞는 제품 2가지를 'product_tool'에서 추천하세요.
출력 형식 예시에 맞게 출력해야 하며, 그 외의 정보는 출력하지 마세요.
```

출력 형식 예시:

- 정확도: ...
- 제품명: ...
- 제품 고유번호: ...
- 추천 이유: ...
- 관련 논문: ...
- 요약: ...

```
""")
```

«수정 후»

Tool 문제 1

Before

product_tool, pdf_tool 둘 다 사용했었음

- product_tool : 코사인 유사도 결과 나온 상위 50개 제품의 정보가 들어가 있는 tool
- pdf_tool : 해당 건강기능식품 관련 논문 정보가 들어가 있는 tool

→ 이 둘을 참조해서 최종 2개의 제품을 추천하는 방식
→ tool을 인식 못하는 문제가 계속 있었음

After

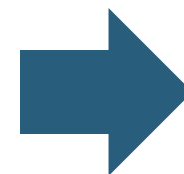
<수정 1>

제품 추천은 코사인 유사도를 이용한 걸로 끝냄 &
RAG에서는 pdf_tool을 이용하여 제품 추천에 대한
근거만 출력하는 방식으로 변경

→ 개선된 점: RAG flow 간단해짐

→ 보완해야 할 점: pdf_tool을 여전히 인식(활용)못함

```
# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[product_tool, pdf_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)
```



```
# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[pdf_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)
```

Tool 문제 2

Before

문제의 원인 파악:

pdf에 있는 논문 내용이 너무 학술적, 일반적인 내용 & 해당 제품의 효능을 직접적으로 다루는 내용이 포함되어 있지 않음

→ pdf_tool을 참조해서 근거를 출력하는 게 불가능

After

<수정 2>

- 논문 pdf_tool 삭제
- 원료명의 효능을 텍스트 파일에 정리
→ txt_tool 생성하여 이것을 참조하도록 함

→ 개선된 점: txt_tool을 잘 인식, 참조하여 근거 출력

프로바이오틱스 (Probiotics)

'살아있는 유익균'으로, 장 건강을 개선하는 데 핵심적인 역할을 합니다.

- * 장내 환경 개선: 유익균을 늘리고 유해균을 억제하여 건강한 장내 미생물 생태계를 만듭니다.
- * 소화 기능 촉진: 음식물 소화를 돕고, 가스나 복부 팽만 같은 불편함을 줄여줍니다.
- * 면역력 증진: 장은 우리 몸 면역 세포의 70% 이상이 모여 있는 곳입니다. 장 건강을 개선해 전반적인 면역 체계를 강화합니다.

← Txt_tool에 들어 있는 '프로바이오틱스'에 대한 효능 정보

↓ 근거 생성

```
{
  "제품명": "MaryRuth Organics USDA Organic Liquid Probiotic | Digestive Health | Gut Health | Probiotics for Women | Probiotics for Men | Probiotics for Kids | Acidophilus Probiotic | Non-GMO | 4C",
  "제품의 효능": "13가지 유기농 프로바이오틱스 균주를 함유한 액상 제품으로, 소화 기능과 장 건강, 면역 기능 개선에 도움을 줄 수 있습니다. 특히 가스, 복부 팽만감 등 소화 불편감을 줄이는 데 효과적입니다.",
  "제품 정보": "USDA 유기농 인증을 받은 비건 액상 프로바이오틱스 제품으로 13가지 고유한 프로바이오틱스 균주로 제조되었습니다. 1세 이상을 위해 제조된 이 제품은 온 가족이 쉽게 복용할 수 있습니다. 독감",
  "근거": "프로바이오틱스는 장내 유익균을 늘리고 유해균을 억제하여 장 건강을 개선하고, 소화 기능을 촉진하며 면역 체계를 강화에 도움을 줍니다.",
  "긍정 리뷰": "소비자들은 소화 개선, 복부 팽만감 감소, 변비 해결 등 장 건강 문제에 대한 효과에 만족했습니다. 일부는 사마귀, 무좀, 습진 등 피부 질환 개선 효과를 경험했습니다. 액상 형태라 아이들에게",
  "주의할 점": "최근 오일 기반으로 변경된 포뮬러의 맛과 질감에 대한 불만이 있으니 주의하세요. 일부 제품에서 곰팡이 발견, 스포이드 파손 등 품질 관리 문제가 보고되었으니 주의하세요. 일부 사용자에게서",
  "요약": "온 가족이 섭취할 수 있는 유기농 액상 프로바이오틱스 제품입니다. 소화 및 장 건강 개선에 대한 긍정적인 후기가 많으나, 최근 변경된 오일 기반 포뮬러의 맛과 질감에 대한 불만과 일부 품질 관리",
}
```


Chapter 5

웹 페이지 구현

A

Lang Chain 기반 웹 페이지

B

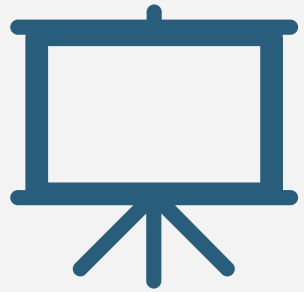
Llama Index 기반 웹 페이지

웹 페이지 구현 예시

건강기능식품 추천 챗봇 UI/UX 플로우



웹 페이지 구성



1

카테고리 페이지

- 카테고리 표시 창
- 원하는 카테고리 선택

2

키워드 페이지

- 주요 키워드 선택 창
- 사용자의 대략적인 니즈 파악



3

입력란 페이지

- 사용자 입력란 창
- 사용자의 세부적인 니즈 파악



4

결과 페이지

- 결과 표시 창
- 전문가 의견 추가
- 사용자 니즈에 맞는 제품 출력



리뷰 출력용 csv 파일 생성

	고유번호	제품명	카테고리	상세정보	긍정_리뷰	부정_리뷰	종합_리뷰
0	10067386235	Newt365	유산균	NaN	섭취 후 변비가 해소되어 매일 화장실을 가게 되었다는 등 배변 활동에 대한 긍정적 ...	광고와 달리 효과를 전혀 보지 못했거나, 처음에는 효과가 있다가 사라졌다는 의견이 ...	제품을 막 받아서 아직 복용 전이거나, 복용한 지 얼마 되지 않아 효과를 기대하며 ...
1	10102052354	Sentroom_man	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명(식약처고시) 비타민A, 비타민D, ...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 복부팽만감과...	제품 효과에 대해 아직 잘 모르겠다는 의견이 다수 있었습니다. 꾸준히 복용하고 있지...
2	10102054620	Sentroom_woman	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명(식약처고시) 비타민A, 비타민D, ...	꾸준히 재구매하는 소비자가 많으며, 피로 개선과 활력 증진에 효과를 느꼈다는 평이 ...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 또한, 배송...	제품을 처음 구매하여 효과를 기대하며 꾸준히 복용해보겠다는 다짐을 남긴 리뷰가 많습...
3	10158604245	Theday	아연	주요 기능성(식약처인증) 면역력 영양소 원료명(식약처고시) 아연, 비타민D, 비타민...	아이들이 맛고맛 젤리 형태라 맛있게 잘 먹고 좋아하며, 스스로 찾아 먹거나 더 달라...	일부 아기들은 젤리 형태나 식감에 익숙하지 않아 먹는 것을 어색해하거나, 씹기 어려...	제품을 구매한 지 얼마 되지 않아 아직 효과는 잘 모르겠지만, 아이의 면역력이 좋아...
4	10164573896	Tenten	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명(식약처고시) 비타민B6, 비타민B2...	아이들이 딸기맛 젤리 같아 거부감 없이 아주 잘 먹는다는 평이 지배적입니다. 다른 ...	약국에서 판매하는 제품과 성분 및 함량이 다른 '텐텐맛' 제품이라 속았다는 불만이 ...	배송이 빠르고 상품을 잘 받았다는 단순 수령 확인 및 감사 인사가 주를 이룹니다. ...

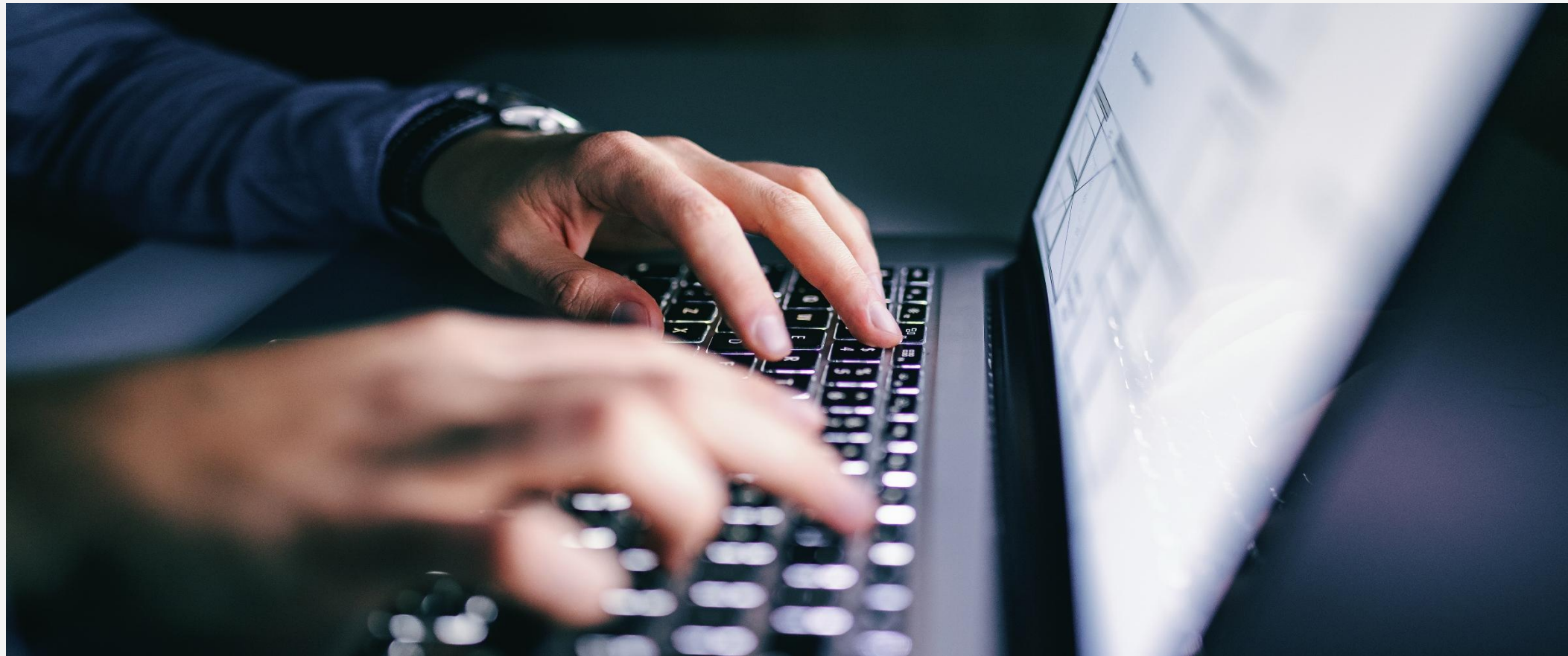
- 웹 페이지 생성 시, 출력되는 리뷰 데이터를 새로 정리
- 구글 제미나이 프로 api를 이용해 리뷰 데이터를 한글로 번역 및 감성별로 정리한 후 리뷰 요약
- 리뷰분석_감성리뷰만_최종본.csv 파일로 생성

임베딩, RAG 파일 생성

```
🐍 create_embeddings.py
🐍 create_rag.py
📄 product_embeddings.joblib
📄 rag_data.joblib
📄 리뷰분석_감성리뷰만_최종본.csv
```

- 웹 페이지 구현 속도를 올리기 위한 조치
- 생성 시, 같은 임베딩, LLM 모델을 이용해 필요한 파일 생성

웹 페이지 실행 : Lang Chain 기반



- 임베딩, RAG 파일을 이용해 웹 페이지 실행하는 코드 구현
- 코사인 유사도 및 RAG 연결을 이용해 리뷰 기반 추천 및 전문가 의견도 반영 가능.

웹 페이지 실행 시 발생한 문제

Before

- 모델의 크기가 너무 큼 -> 로컬 환경에서 돌아가기엔 버겁다는 문제 발생
- 전문가 의견 생성 시, Lang Chain 기반 rag의 경우 pdf 파싱이 원활히 되지 않음

After

- rag 연결을 Lang Chain 기반 보다는 Llama Index 기반의 rag로 교체하기로 결정
- 단순 llm 호출 보다는 에이전트와 tool을 활용하여 전문가 의견 출력 성능을 높이려 시도

전문가 의견

아연(Zinc)은 면역 체계 강화와 세포 분열에 필수적인 미네랄로, 적정 용량에서는 부작용이 거의 없으며, 과다 섭취 시 위장 장애나 구리 흡수 방해 등의 문제가 발생할 수 있습니다.

참고 문서

- **** (페이지: -1)
- **** (페이지: -1)
- **** (페이지: -1)

리뷰 출력용 csv 파일 재생성

고유번호	제품명	카 테 고 리	상세정보	긍정리뷰	부정리뷰	중립리뷰	긍정_청크	긍정리뷰_임베딩	
0	10067386235	Newt365	유산균	NaN	섭취 후 변비가 해소되어 매일 화장실을 가게 되었다는 등 배변 활동에 대한 긍정적 ...	광고와 달리 효과를 전혀 보지 못했거나, 처음에는 효과가 있다가 사라졌다는 의견이 ...	제품을 막 받아서 아직 복용 전이거나, 복용한 지 얼마 되지 않아 효과를 기대하며 ...	섭취 후 변비가 해소되어 매일 화장실을 가게 되었다는 등 배변 활동에 대한 긍정적 ...	[0.10404543578624725, -0.11060380935668945, -0...
1	10102052354	Sentroom_man	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명 (식약처고시) 비타민A, 비타민D, ...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 복부 팽만감과...	제품 효과에 대해 아직 잘 모르겠다는 의견이 다수 있었습니다. 꾸준히 복용하고 있지...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	[-0.06720072776079178, 0.08641089498996735, -0...
2	10102052354	Sentroom_man	비타민	주요 기능성(식약처인증) 영양보충 영양소 원료명 (식약처고시) 비타민A, 비타민D, ...	많은 소비자들이 피로 회복과 컨디션 개선 효과를 경험했으며, 먹은 날과 안 먹은 날...	일부 소비자들은 알약의 크기가 커서 목 넘김이 불편하다고 지적했습니다. 복부 팽만감과...	제품 효과에 대해 아직 잘 모르겠다는 의견이 다수 있었습니다. 꾸준히 복용하고 있지...	저렴한 가격과 브랜드에 대한 신뢰도 긍정적인 요인으로 언급되었습니다.	[-0.2792225182056427, 0.12212540209293365, -0....

- 리뷰분석_감성리뷰만_최종본.csv 파일에 긍정 리뷰를 임베딩한 값을 새 열로 추가
- reviews_chunked_embeddings4.csv 파일로 새로 생성 후 사용

웹 페이지 실행 : Llama Index 기반



에이전트 교체

Before

ReActAgent 로 스트리밍 실행 시, tool 오류 발생
전문가 의견 산출 실패 지속.

After

QueryEngine을 사용
중간 과정 없이 LLM에 직접 프롬프트를 전달해
원하는 답변을 생성, 안정성 확보



J son 파싱

Before

정규식으로 j son 블록만 추출하는 로직을
추가했으나, LLM의 응답은 항상 예상대로 나오지
않을 가능성이 높음

After

extract_json_from_text 함수를 만들어 이
문제를 해결할 수 있는 매우 정교한 파싱 로직을
생성



UI 통합

Before

'추천 제품'과 '전문가 의견' 섹션이 분리된 상태 ->
이로 인해 한 제품에 대한 정보가 중복되는 문제 발견

After

두 섹션을 '**추천 제품 및 전문가 의견**'이라는 하나의
섹션으로 통합 및 프롬프트 수정

웹 페이지 실행 : Llama Index 기반

```
# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[pdf_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)
```



```
# 1. QueryEngine 직접 호출
try:
    response = txt_engine.query(user_prompt)
    response_str = str(response)
except Exception as e:
    st.error(f"쿼리 엔진 호출 중 오류가 발생했습니다: {e}")
    return None
```

웹 페이지 실행 : Llama Index 기반

```
# ReActAgent 생성
healthcare_agent = ReActAgent(
    tools=[product_tool, txt_tool],
    llm=Settings.llm,
    system_prompt=react_system_prompt,
    verbose=True
)

response = await healthcare_agent.run(user_msg=f"""당신은 사용자의 요구에 가장 알맞는 제품을 추천하는 친절하고 유능한 건강기능식품 AI 비서입니다.
주어진 도구를 사용하여 질문에 대한 정보를 찾아야 합니다. 'pdf_tool'를 사용하세요.
'product_text'에 있는 상세정보에 있는 영양소 원료명을 'pdf_tool'을 참조하여 제일 사용자의 요구에 맞는 제품 2가지를 'product_tool'에서 추천하
출력 형식 예시에 맞게 출력해야 하며, 그 외의 정보는 출력하지 마세요.

출력 형식 예시:
- 정확도: ...
- 제품명: ...
- 제품 고유번호: ...
- 추천 이유: ...
- 관련 논문: ...
- 요약: ...
""")
```

```
# 2. JSON 파싱
json_str = extract_json_from_text(response_str)

if json_str:
    try:
        return json.loads(json_str)
    except Exception:
        pass

# 최종적으로 전체 문자열에서 파싱 시도
try:
    return json.loads(response_str)
except (json.JSONDecodeError, SyntaxError) as e:
    st.error("LLM 응답에서 JSON을 파싱하지 못했습니다.")
    st.json(response_str)
    return None
```

웹 페이지 실행 : Llama Index 기반

```
st.markdown("### 🧑‍⚕️ 전문가 의견")
with st.container(border=True):
    if st.session_state.get('rag_nodes') and Settings.llm is not None:
        selected_supplement_name = st.session_state.get('selected_supplement_name', '건강식품')
        selected_keywords = st.session_state.get('selected_keywords', [])
        user_free_text = st.session_state.get('user_free_text', '')
        with st.spinner("PDF 문서를 기반으로 전문가 의견을 생성 중..."):
            rag_query = f"{selected_supplement_name}의 효과에 대해 알려주세요. {' '.join(selected_keywords)}"
            rag_result = answer_query(rag_query)
            if '전문가의 의견' in rag_result and rag_result['전문가의 의견'] != '정보 부족':
                st.markdown(rag_result['전문가의 의견'])
            else:
                st.markdown("선택하신 조건에 대한 전문가 의견을 찾을 수 없습니다.")
            if '참고 문서' in rag_result and rag_result['참고 문서']:
                st.markdown("#### 참고 문서")
                # doc이 딕셔너리인지 확인하고 get()을 사용
                for doc in rag_result['참고 문서']:
                    if isinstance(doc, dict):
                        title = doc.get('title', '제목 없음')
                        page = doc.get('page', '?')
                        st.markdown(f"- **{title}** (페이지: {page})")
                    else:
                        # doc이 딕셔너리가 아닐 경우를 대비한 처리
                        st.markdown(f"- 참고 문서를 불러오지 못했습니다. (데이터 형식 오류)")
            else:
                st.warning("RAG 시스템이 초기화되지 않았습니다. PDF 파일과 라이브러리를 확인해주세요.")

st.markdown("### 🛒 추천 제품")
with st.container(border=True):
```

Windows 정

```
st.markdown("### 🛒 추천 제품 및 전문가 의견") # 제목 변경
with st.container():
    st.markdown("<div style='border:1px solid #eee; padding:12px; border-radius:8px'>", unsafe_
    rag_result = st.session_state.get('rag_result', None)
    if rag_result and isinstance(rag_result, list):
        for product_data in rag_result:
            if '제품명' in product_data:
                st.markdown(f"#### {product_data.get('제품명', '제품명 없음')}")
                st.markdown(f"- **제품의 효능:** {product_data.get('제품의 효능', '정보 없음')}")
                st.markdown(f"- **제품 정보:** {product_data.get('제품 정보', '정보 없음')}")
                st.markdown(f"- **근거:** {product_data.get('근거', '출처 없음')}")
                st.markdown(f"- **요약:** {product_data.get('요약', '정보 없음')}")
                st.markdown("---")
                st.markdown("##### 📄 참고할만한 리뷰들")
                st.markdown(f"- **긍정 리뷰:** {product_data.get('긍정 리뷰', '정보 없음')}")
                st.markdown(f"- **부정 리뷰:** {product_data.get('부정 리뷰', '정보 없음')}")
                st.markdown(f"- **중립 리뷰:** {product_data.get('중립 리뷰', '정보 없음')}")
                st.markdown(f"- **주의할 점:**")
                warnings = product_data.get('주의할 점', [])
                if isinstance(warnings, list):
                    for warning in warnings:
                        st.markdown(f"- {warning}")
                else:
                    st.markdown(f"- {warnings}")
            st.divider()
```

Windows 정품

웹 페이지 실행 구현

AI 헬스 메이트 시연 영상

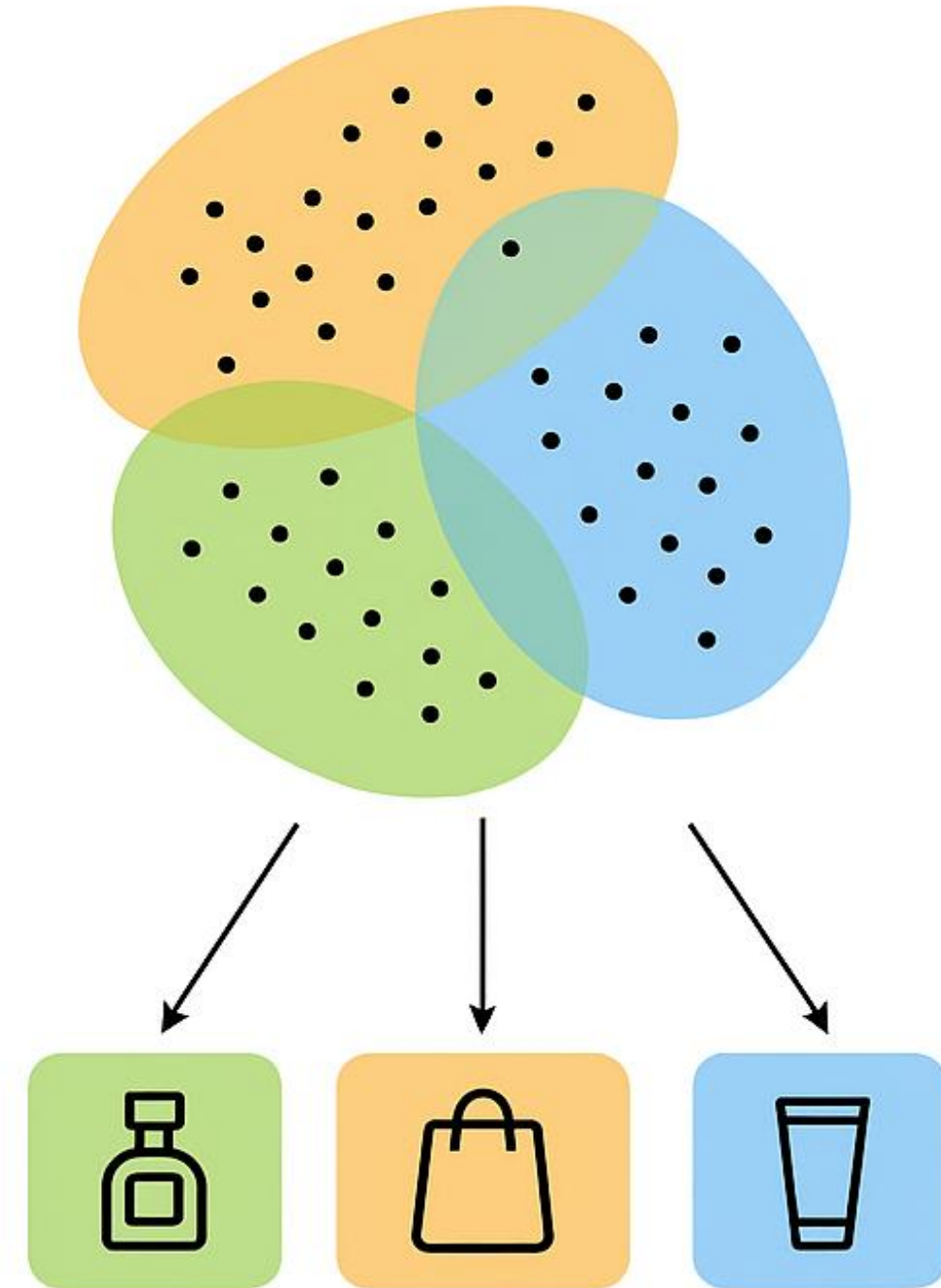
차후 개선사항

1. 유사도 계산 성능 향상

- 단순 코사인 유사도 → Sentence-BERT, SimCSE 등 임베딩 기반 기법 적용
- FAISS 등 효율적 최근법 탐색 도입
→ 대규모 데이터에서도 빠른 검색

2. 개인화 추천 고도화

- GMM 기반 Soft Clustering → 사용자가 여러 관심사 그룹에 속할 수 있도록 모델링
- 다중 관심사 반영 → 보다 정교한 맞춤형 제품 추천 가능





감사합니다

여기까지 2팀의 최종 프로젝트를 마치겠습니다.

2팀

조장: 김성한
조원: 김민경, 김태윤